Progetto S2/L5

Josh V. E. Abanico
CS0525

5 dicembre 2025

Indice

1 Obiettivi del progetto

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica. Dato il codice si richiede allo studente di:

1. Capire cosa fa il programma **senza eseguirlo**.
2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

```
1   import datetime
2
3   while True
4       comando_utente = input("Cosa vuoi sapere? ")
5       if comando_utente == "esci":
6           print("Arrivederci!")
7       break
8       else:
9           print(assistente_virtuale(comando_utente))
10
11  def assistente_virtuale(comando):
12      if comando == "Qual è la data di oggi?":
13          oggi = datetime.datetoday()
14          risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
15      elif comando == "Che ore sono?":
16          ora_attuale = datetime.datetime.now().time()
17          risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
18      elif comando == "Come ti chiami?":
19          risposta = "Mi chiamo Assistente Virtuale"
20      else:
21          risposta = "Non ho capito la tua domanda."
22      return risposta
23
```

Figura 1: Codice sorgente

2 Il codice

Il codice Python è pensato per sviluppare un assistente virtuale che chiede all'utente 'Cosa vuoi sapere?'. È possibile richiedere la data di oggi, l'ora e il nome dell'assistente, oppure chiudere il programma digitando 'esci'. Al momento, sono presenti alcuni errori sintattici e logici che elencherò a breve.

3 Errori

3.1 Errori sintattici

Questi errori violano la grammatica del linguaggio Python e impediscono l'avvio dello script.

- Mancata formattazione del ciclo `while` **(Riga 3)**: L'istruzione `while` è priva dei due punti finali (`:`), necessari per definire l'inizio del blocco di codice.
- Indentazione errata (**Riga 7**): Il comando `break` non rispetta l'indentazione corretta rispetto al blocco in cui è contenuto, generando un `IndentationError`.

3.2 Errori logici

Questi errori si verificano durante l'esecuzione del programma, causandone l'interruzione improvvisa (crash).

- Ordine di definizione errato (**Riga 9**): Viene effettuata una chiamata a una funzione prima che questa sia stata effettivamente definita nel codice (`NameError`). In Python, la definizione (`def`) deve precedere l'utilizzo.
- Metodo inesistente (Libreria `datetime`): Viene invocato il metodo `.datetoday()`, che non appartiene alla libreria standard `datetime`. La sintassi corretta dovrebbe essere `.datetime.datetime.today()`.

```
1   import datetime
2
3   while True
4       comando_utente = input("Cosa vuoi sapere? ")
5       if comando_utente == "esci":
6           print("Arrivederci!")
7       break
8       else:
9           print(assistente_virtuale(comando_utente))
10
11  def assistente_virtuale(comando):
12      if comando == "Qual è la data di oggi?":
13          oggi = datetime.datetoday()
14          risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
15      elif comando == "Che ore sono?":
16          ora_attuale = datetime.datetime.now().time()
17          risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
18      elif comando == "Come ti chiami?":
19          risposta = "Mi chiamo Assistente Virtuale"
20      else:
21          risposta = "Non ho capito la tua domanda."
22      return risposta
23
```

Figura 2: Analisi errori codice sorgente

4 Difetti di progettazione

4.1 Assenza di istruzioni operative e menu comandi

All'avvio, il programma presenta un prompt generico (`"Cosa vuoi sapere?"`) senza fornire all'utente un elenco delle funzionalità disponibili. Questo costringe l'utente a tentare di indovinare i comandi validi, rendendo l'interazione frustrante e poco intuitiva. Manca un messaggio di benvenuto o un menu che espliciti le opzioni supportate (es. Data, Ora, Nome).

4.2 Gestione rigida dell'input utente (Case Sensitivity)

Il codice attuale effettua un confronto diretto tra l'input dell'utente e le stringhe di comando (es. `if risposta == "Che ore sono?"`). Questo approccio è case-sensitive (sensibile alle maiuscole/minuscole): se l'utente digita `"ore"` o `"ORE"` invece di `Ore`, il programma non riconosce il comando.

5 Soluzione

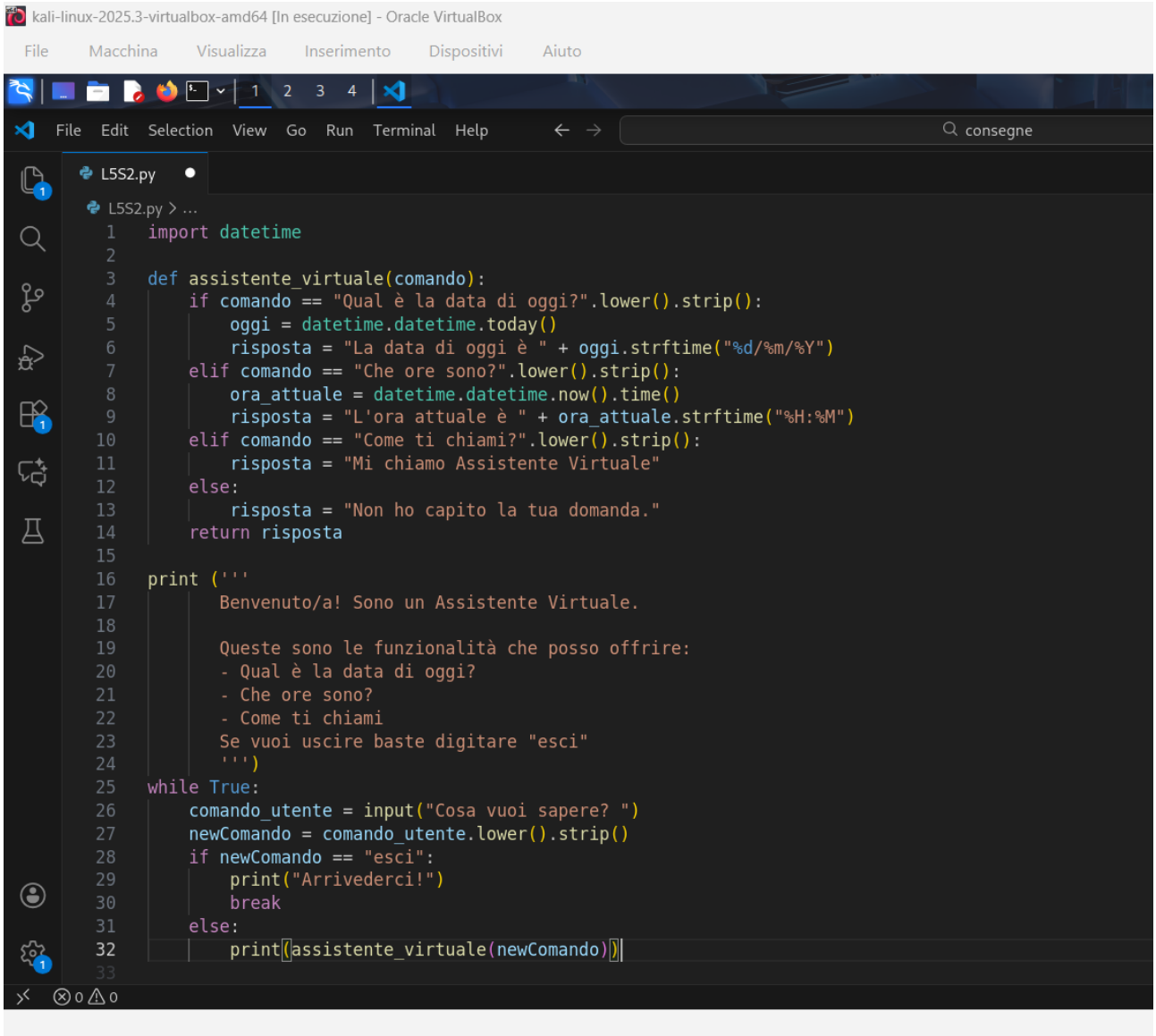

Figura 3: Correzione del codice sorgente

1. Ho messo la funzione `assistente_virtuale(comando)` sopra il blocco `while` perché così nella **Riga 32** la funzione viene definita.

2. Ho corretto il metodo `.datetoday()` con `.datetime.datetime.today()` nella **Riga 5**

3. All'interno della funzione assistente_virtuale(comando) **(Riga 3-14)** negli `if - elif` ho messo dopo la stringa di comando un `.lower()` per mettere tutte le stringhe in lowercase e `.strip()` di cancellare gli spazi prima o dopo la stringa prima di effettuare il confronto.

4. Il confronto lo deve fare con il comando fatto dall'utente e lo troviamo all'interno del blocco `while` specificatamente nella **Riga 27**. Qui ho dovuto trasformare la variabile `comando_utente` con i metodi `.lower()` e `.strip()` mettendolo in una nuova variabile `newComando` così da riuscire a confrontarli con le stringhe di comando nella funzione assistente_virtuale(comando)

5. Per essere più chiaro all'utente, ho inserito un menu di benvenuto che lo puoi trovare nella **Riga 16-24**

6 Test

Ho provato a far partire il programma, con vari input da testare la case-sensitivity del programma e il menu iniziale è molto chiaro e diretto.

```
┌──(kali㉿kali)-[~/Desktop/Python-programming/consegne]
└─$ python L5S2.py

      Benvenuto/a! Sono un Assistente Virtuale.

      Queste sono le funzionalità che posso offrire:
      - Qual è la data di oggi?
      - Che ore sono?
      - Come ti chiami
      Se vuoi uscire baste digitare "esci"

Cosa vuoi sapere? Qual è la data di oggi?
La data di oggi è 05/12/2025
Cosa vuoi sapere? Che ORE sono?
L'ora attuale è 07:24
Cosa vuoi sapere? CoMe Ti ChIaMi?
Mi chiamo Assistente Virtuale
Cosa vuoi sapere? ESCI
Arrivederci!

┌──(kali㉿kali)-[~/Desktop/Python-programming/consegne]
└─$
```

Figura 4: Test del codice